

Assignment #9: dfs, bfs, & dp

Updated 2107 GMT+8 Nov 19, 2024

2024 fall, Compiled by 昂奕 化学与分子工程学院

说明:

- 1) 请把每个题目解题思路（可选），源码Python, 或者C++（已经在Codeforces/Openjudge上AC），截图（包含Accepted），填写到下面作业模版中（推荐使用 typora <https://typoraio.cn>，或者用word）。AC 或者没有AC，都请标上每个题目大致花费时间。
- 2) 提交时候先提交pdf文件，再把md或者doc文件上传到右侧“作业评论”。Canvas需要有同学清晰头像、提交文件有pdf、“作业评论”区有上传的md或者doc附件。
- 3) 如果不能在截止前提交作业，请写明原因。

1. 题目

18160: 最大连通域面积

dfs similar, <http://cs101.openjudge.cn/practice/18160>

思路: dfs, 但是@lru_cache在第二遍大循环的时候如果传入相同的参数不会再执行函数体导致结果出错

代码:

```
import sys
sys.setrecursionlimit(1<<30)

def dfs(x,y):
    global S
    d=[(-1,-1),(-1,0),(-1,1),(0,-1),(0,1),(1,-1),(1,0),(1,1)]
    mat[x][y]='.'
    for dx,dy in d:
        nx,ny=x+dx,y+dy
        if 0<=nx<len(mat) and 0<=ny<len(mat[0]):
            if mat[nx][ny]=='W':
                S+=1
                dfs(nx,ny)

t=int(input())
for _ in range(t):
    n,m=map(int,input().split())
    mat=[]
    for i in range(n):
        mat.append(list(input()))
    maxs=0
    for i in range(n):
        for j in range(m):
            if mat[i][j]=='W':
                S=1
                dfs(i,j)
            maxs=max(S,maxs)
```

状态: Accepted

源代码

```
import sys
sys.setrecursionlimit(1<<30)

def dfs(x,y):
    global S
    d=[(-1,-1),(-1,0),(-1,1),(0,-1),(0,1),(1,-1),(1,0),(1,1)]
    mat[x][y]='.'
    for dx,dy in d:
        nx,ny=x+dx,y+dy
```

基本信息

#: 47306010
题目: 18160
提交人: 24n2400011782
内存: 3704kB
时间: 109ms
语言: Python3
提交时间: 2024-11-21 15:30:52

19930: 寻宝

bfs, <http://cs101.openjudge.cn/practice/19930>

思路:
模板题
代码:

```
from collections import deque

def bfs(x,y):
    q=deque([(x,y,0)])
    d=[(-1,0),(0,1),(0,-1),(1,0)]
    while q:
        x,y,cnt=q.popleft()
        if mat[x][y]==1:
            return cnt
        mat[x][y]=2
        for dx,dy in d:
            nx,ny=x+dx,y+dy
            if 0<=nx<len(mat) and 0<=ny<len(mat[0]):
                if mat[nx][ny]==0:
                    q.append((nx,ny,cnt+1))
                elif mat[nx][ny]==1:
                    return cnt+1
        else: return('NO')
n,m=map(int,input().split())
mat=[]
for _ in range(n):
    mat.append(list(map(int,input().split())))
print(bfs(0,0))
```

代码运行截图 ==（至少包含有"Accepted"）==

04123: 马走日

dfs, <http://cs101.openjudge.cn/practice/04123>

思路:
dfs就完了
代码:

```
#pylint:skip-file
def dfs(x,y,cnt):
    global n,m,path,mat
    d=[(-2,-1),(-2,1),(2,-1),(2,1),(-1,-2),(-1,2),(1,-2),(1,2)]
    if cnt==n*m:
        path+=1
        return
    for dx,dy in d:
```

```

nx,ny=x+dx,y+dy
if 0<=nx<n and 0<=ny<m and mat[nx][ny]==0:

    mat[x][y]=1
    dfs(nx,ny,cnt+1)
    mat[x][y]=0
t=int(input())
for i in range(t):
    n,m,x,y=map(int,input().split())
    path=0
    mat=[[0]*m for _ in range(n)]
    dfs(x,y,1)
    print(path)

```

代码运行截图（至少包含有"Accepted"）

状态: Accepted

源代码

```

#pylint:skip-file
def dfs(x,y,cnt):
    global n,m,path,mat
    d=[(-2,-1),(-2,1),(2,-1),(2,1),(-1,-2),(-1,2),(1,-2),(1,2)]
    if cnt==n*m:
        path+=1
        return
    for dx,dy in d:
        nx,ny=x+dx,y+dy
        if 0<=nx<n and 0<=ny<m and mat[nx][ny]==0:

            mat[x][y]=1
            dfs(nx,ny,cnt+1)
            mat[x][y]=0
t=int(input())
for i in range(t):
    n,m,x,y=map(int,input().split())
    path=0
    mat=[[0]*m for _ in range(n)]
    dfs(x,y,1)
    print(path)

```

sy316: 矩阵最大权值路径

dfs, <https://sunnywhy.com/sfbj/8/1/316>

思路:

dfs.我把当成上一题只能往右往下走WA了三次...

代码:

```

        max_path=path[:]
    path.pop(-1)
    w-=mat[x][y]
    return
    for dx,dy in d:
        nx,ny=x+dx,y+dy
        if 1<=nx<=n and 1<=ny<=m and not visited[nx][ny]:
            visited[x][y]=1
            dfs(nx,ny)
            visited[x][y]=0
    path.pop(-1)
    w-=mat[x][y]
n,m=map(int,input().split())
mat=[[0]*(1+m)]
for i in range(n):
    mat.append([0]+list(map(int,input().split())))
visited=[[0]*(1+m) for _ in range(n+1)]
dfs(1,1)
for i in max_path:
    print(*i,sep=' ')

```

代码书写



Python ▾

```
1 path=[]
2 max_path=[]
3 w=0
4 wmax=-999999
5 def dfs(x,y):
6     global n,m,w,wmax,max_path
7     w+=mat[x][y]
8     path.append((x,y))
9     d=[(1,0),(0,1),(-1,0),(0,-1)]
10    if (x,y)==(n,m):
11        #print(f'终点\npath={path},\nw={w}')
12        if w>wmax:
13            wmax=w
14            max_path=path[:]
15    path.pop(-1)
```

测试输入

提交结果

历史提交

完美通过[查看题解](#)**100% 数据通过测试****运行时长: 0 ms**[收起面板](#)

运行



提交

LeetCode62.不同路径dp, <https://leetcode.cn/problems/unique-paths/>

思路:

dp,创建一个和网格同样大小的矩阵, 里面存的是相应的每个位置有多少种不同的方法可到。

 $dp[x][y]=dp[x-1][y]+dp[x][y-1]$

代码:

```
class Solution:
    def uniquePaths(self, m: int, n: int) -> int:
        dp=[0]*(n+1) for _ in range(m+1)
        dp[0][1]=1
        for x in range(1,m+1):
            for y in range(1,n+1):
                dp[x][y]=dp[x-1][y]+dp[x][y-1]
        return(dp[m][n])
```

代码运行截图（至少包含有"Accepted"）

[题目描述](#)
[通过](#)
[题解](#)
[提交记录](#)

- 全部提交记录

通过

12 提交于 2024.11.23 16:35

官方题解

写题解



「感恩季」限时福利！
买 1 得 2 同时可享力扣 365 天 Plus 会员 + 热门平台会员权益。

→

执行用时分布

</> 代码

Python3 智能模式

```
1 class Solution:
2     def uniquePaths(self, m: int, n: int) -> int:
3         dp=[0]*(n+1) for _ in range(m+1)
4         dp[0][1]=1
5         for x in range(1,m+1):
6             for y in range(1,n+1):
7                 dp[x][y]=dp[x-1][y]+dp[x][y-1]
8         return(dp[m][n])
```

sy358: 受到祝福的平方

dfs, dp, <https://sunnywhy.com/sfbj/8/3/539>

思路:

dfs。

代码:

```
import math
blessed=False
def dfs(num):
    global blessed
    for i in range(1,len(num)+1):
        rt=math.sqrt(int(num[0:i]))
        if rt%1==0 and rt!=0:
            if i==len(num):
                blessed=True
                return
            else:
                dfs(num[i:])
    else:
        return
n=input()
dfs(n)
if blessed:print('Yes')
else:print('No')
```

题目题解

受到祝福的平方

通过数 256 提交数 518 难度 中等 显示标签 ☆

题目描述

在小元的世界里，任何人出生后会被世界分配一个随机ID，如果ID在被切割后，即ID满足按照从左至右顺序分割，且分割出来的数字都是某一个正整数的平方，分割时可以包括前导0，那么他就被这个世界祝福，最后获得快乐的数量和质量都比不满足这样的ID的人多的多。

令ID为A，且A是一个正整数，取值范围为 $1 \leq A \leq 10^9$ ，问A是否是一个被受到祝福的ID。

比如A = 8194时，它是一个被受到祝福的ID，因为他可以被分割为 $\{81, 9, 4\} = \{9^2, 3^2, 2^2\}$ ；

代码书写

```
1 import math
2 blessed=False
3 def dfs(num):
4     global blessed
5     for i in range(1,len(num)+1):
6         rt=math.sqrt(int(num[0:i]))
7         if rt%1==0 and rt!=0:
8             if i==len(num):
9                 blessed=True
10                return
11            else:
12                dfs(num[i:])
13        else:
```

测试输入 提交结果 历史提交

完美通过

100% 数据通过测试

运行时长: 0 ms

查看题解

2. 学习总结和收获

如果作业题目简单，有否额外练习题目，比如：OJ“计概2024fall每日选做”、CF、LeetCode、洛谷等网站题目。

正在补OJ选做题...dp还是还是不太会